

REMARKS

The above-identified patent application has been reviewed in light of the Examiner's Action dated October 16, 2006. Claims 1, 6, 8, 10, 11, 14 and 15 have been amended, without intending to abandon or to dedicate to the public any patentable subject matter. No claims have been canceled. Accordingly, Claims 1-21 are now pending. As set forth herein, reconsideration and withdrawal of objections to and rejections of the claims are respectfully requested.

The drawings stand objected to because the drawings previously submitted in connection with a Notice to File Corrected Application Papers were not labeled as "Replacement Sheet." In order to streamline prosecution, Applicant has acceded to the Examiner's request, and has re-submitted formalized drawings concurrently herewith that are identical to those previously submitted, except that they now incorporate the label "Replacement Sheet." Accordingly, the Examiner's stated basis for objecting to the drawings has been removed, and reconsideration and withdrawal of the objection is requested.

Claims 8 and 15 stand objected to because of noted informalities. In the amendments set forth above, changes suggested by the Examiner have been incorporated. Accordingly, it is submitted that the objections to Claims 8 and 15 should be reconsidered and withdrawn.

Claims 14 and 15 stand rejected under 35 U.S.C. § 112, first paragraph, as containing subject matter that was not described in the specification. In the amendment set forth above, Claims 14 and 15 have been amended to recite "determining a number of bytes." This aspect of the claimed invention is supported in the specification, for example at page 8, lines 3-18. Accordingly, it is submitted that the rejections of Claims 14 and 15 as failing to comply with the

written description requirement should be reconsidered and withdrawn.

Claims 1-9 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. In the amendments set forth above, Claim 1 has been amended to initially recite “an update table” and “a memory.” Accordingly, it is submitted that the basis for this rejection has been removed. Therefore, the rejections of claims 1-9 as indefinite should be reconsidered and withdrawn.

Claims 10-17 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. In the amendments set forth above, structural and functional inter-relationships have been further specified. In view of these amendments, it is submitted that the rejections of Claims 10-17 under 35 U.S.C. § 101 should be reconsidered and withdrawn.

Claims 1-5, 7, 10-13, and 16-19 stand rejected under 35 U.S.C. § 102 as being anticipated by U.S. Patent No. 5,619,698 to Lillich et al. (“Lillich”). In addition, Claims 6, 8, 9, 14, 15, 20, and 21 stand rejected under 35 U.S.C. § 103 as being obvious over Lillich in view of U.S. Patent Application Publication No. 2004/0107416 to Buban et al. (“Buban”), Lillich in view of U.S. Patent Application Publication No. 2003/0167463 to Munsil et al. (“Munsil”), or U.S. Patent Application Publication No. 2002/0152455 to Hundt et al (“Hundt”) in view of Buban. In order for a rejection under 35 U.S.C. § 102 to be proper, each and every element as set forth in the claim must be found, either expressly or inherently described, in a single prior art reference (MPEP § 2131). In order to establish a *prima facie* case of obviousness under §103, there must be some suggestion or motivation to modify the reference or to combine the reference teachings, there must be a reasonable expectation of success, and the prior art reference or references must

teach or suggest all of the claim limitations (MPEP § 2143). However, all of the claim elements cannot be found in the cited references, whether those references are considered alone or in combination. Accordingly, reconsideration and withdrawal of the rejections of the claims as anticipated by or obvious in view of the cited references are respectfully requested.

The present invention is generally directed to updating running processes. As set forth in at least some of the pending claims, a jump instruction and an address of an update table is injected at a location in a memory containing a first instruction of a first replaced function. That is, the jump instruction is placed in the body of executable code. This is in contrast to references cited by the Office Action, in which an address in a jump table that previously referred to an address of a first instruction of executable code associated with a replaced body of code is replaced by an address of a first instruction associated with a replacement body of code. In accordance with other claims, a jump instruction and an address of a new function is injected in a running executable program for execution by a processor in place of a replaced function. As noted above, this is in contrast to cited references in which a replacement function is called in place of a previously active function by replacing an address in a jump table associated with the previous function with the address of the replacement function. Still other embodiments of the present invention provide tools for updating running processes in which an address of a function to be replaced is replaced with an address of the replacement function in response to the instruction pointer being at least a predetermined distance from the address of the replacement function.

The Lillich reference is generally directed to a method and apparatus for patching

operating systems. In the background discussion of Lillich, which has been cited by the Office Action, a client call made to a replaced function is re-directed to the patch comprising the computer code replacing the called function. More particularly, with respect to a system routine that is not patched, upon reaching an ATRAP instruction, a trap dispatcher evaluates the ATRAP instruction and looks up the corresponding system routine in the trap table. (Lillich, column 3, lines 6-25.) In order to replace existing system routines, individual entries in the trap table can be changed to point to addresses other than the ROM addresses. (Lillich, column 3, lines 26-30.) Accordingly, the discussion in Lillich cited by the Office Action discusses replacing in a trap table the address of the first line of code of a replacement function. Therefore, this discussion in Lillich does not teach, suggest, or describe injecting a jump instruction at a location in a memory containing a first instruction of a first replaced function.

Accordingly, at least the following elements of the independent claims that stand rejected as being anticipated by Lillich, indicated below in italicized text, cannot be found in that reference:

1. A method for updating a running process, comprising:

 allocating in executable program code text first memory space operable to receive new program instructions comprising at least a first updated function;

 allocating in executable program code text second memory space operable to receive address information related to said new program instructions;

 running said executable program code;

 stopping execution of said executable program code;

injecting a jump instruction and an address of an update table at a location in a memory containing a first instruction of a first replaced function, wherein said address of said update table contains an address of a first instruction of said first updated function; and

resuming execution of said executable program code, wherein said first updated function is called in place of said first function, and wherein said executable code is updated in said memory.

10. A computer implemented method, the method comprising:

receiving information identifying:

a running executable program to be patched; and

a function to be replaced;

accessing a symbol table in a memory for said executable program to be patched;

obtaining from said symbol table an address of said function to be replaced;

stopping execution by a processor of said running executable program to be patched;

injecting in said running executable program to be patched at a location in said memory containing a first instruction of said function to be replaced a jump instruction and an address of a new function, wherein said new function is executed by said processor in place of said function to be replaced, and wherein a patched version of said executable program is created in said memory; and

resuming execution of said executable program by said processor, wherein said patched version of said executable program is executed by said processor.

18. A system for updating executable program code, comprising:

- means for receiving information identifying existing executable program code;
- means for receiving information identifying a function to be replaced;
- means for stopping execution of said existing executable program code;
- means for inserting a jump code and an address associated with a replacement function in place of an address of said function to be replaced in said existing executable program code, wherein updated executable program code is created;* and
- means for resuming execution of said executable program code, wherein said updated executable program code is executed.

Therefore, Independent Claims 1, 10, and 18 and Dependent Claims 2-5, 7, 11-13, 16-17 and 19 are not anticipated by Lillich, and rejections of these claims should be reconsidered and withdrawn.

The Buban reference has been cited in combination with Lillich in connection with the rejections of Claims 6, 8, 14 and 15. In particular, the Office Action acknowledges that Lillich does not describe determining a distance between the position within code text at which execution of a program is stopped and an address of a function is to be updated. It is noted that the Buban reference, which is cited with respect to determining a distance within program code, does not teach, suggest or describe those aspects of the independent claims missing from Lillich as discussed above. Accordingly, Claims 6, 8, 14 and 15 should be allowed for at least the same reasons that the Independent Claims from which they depend from are allowable. In addition, the portion of the discussion in Buban cited in the Office Action does not discuss a pre-determined

distance between memory addresses and populating an update table with the address of a first updated function in response to determining that the first distance is at least the pre-determined amount. Instead, Buban discusses preferably limiting the size of the instruction being updated to the processor's smallest unit of atomically replaceable memory (*e.g.* 8 bytes). (Buban, paragraph 46.) Therefore, the Buban reference does not teach, suggest, or disclose such aspects of Claims 6, 8, 14 and 15, and the rejections of these claims should be reconsidered and withdrawn.

Claim 9 stands rejected as being unpatentable over Lillich in view of Munsil. In particular, the Office Action acknowledges that Lillich does not disclose a second memory space that is read-only memory space. However, it is noted that Claim 9 should be allowed for at least the same reasons that Claim 1 is allowable. Accordingly, for at least these reasons, the rejection of Claim 9 should be reconsidered and withdrawn.

Claims 20 and 21 stand rejected as being unpatentable over Hundt in view of Buban. In general, Claim 20 is directed to a system for updating and executing program code. The system includes a signal handler tool operable to replace a memory and address of the function to be replaced with the address of a replacement function. Moreover, such a replacement is performed in response to the instruction pointer being at least a pre-determined distance from the address for the replacement function. These aspects of the system of Claim 20 cannot be found in the cited references.

The Hundt reference is generally directed to dynamic instrumentation of an executable program. In general, Hundt discusses using original functions to generate instrumented functions, and executing the instrumented functions instead of the original functions. (Hundt,


Application No. 10/650,257

paragraph 18.) As acknowledged in the Office Action, there is no disclosure in Hundt related to making such a replacement in response to the position of the instruction pointer being at least a pre-determined distance from the address of the replacement function. For such disclosure, the Office Action cites Buban. However, as previously noted, Buban in fact contains no teaching, suggestion or disclosure of such a pre-determined distance. Accordingly, Claim 20 and Dependent Claim 21 should be allowed.

The application now appearing to be in condition for allowance, early notification of same is respectfully requested. The Examiner is invited to contact the undersigned by telephone if doing so would be of assistance.

Respectfully submitted,

SHERIDAN ROSS P.C.

By: 
Bradley M. Knepper
Registration No. 44,189
1560 Broadway, Suite 1200
Denver, Colorado 80202-5141
(303) 863-9700

Date: July 16, 2007